

In the claims:

Please amend the claims as follows:

*b1* 1. (Currently Amended): A method for executing a graph having vertices representing components and links between components indicating flows of data between such components, the graph having components with parameters, including:

- (a) retrieving a runtime parameter for the graph at runtime execution of the graph, the runtime parameter having a value defined as determinable at runtime execution of the graph;
- (b) determining whether the value for the runtime parameter is to be provided by user input;
- (c) displaying a prompt to a user for receiving user input for every runtime parameter so determined;
- (d) determining a first final parameter value based on user response to such prompt; and
- (e) executing the graph using the first final parameter value as the value for the runtime parameter.

2. (Original): The method of claim 1, further including:

- (a) determining whether the value for the runtime parameter is to be externally supplied programmatically; and
- (b) retrieving any externally supplied value for every runtime parameter so determined;
- (c) determining a second final parameter value based on such externally supplied value; and
- (d) executing the graph using the second final parameter value as the value for the runtime parameter.

3. (Original): The method of claim 1, further including providing an interface which permits designating a parameter of a graph component as a runtime parameter.

4. (Original): The method of claim 1, wherein determining the first final parameter value includes evaluating an expression.

5. (Original): The method of claim 4, wherein the expression computes metadata.

6. (Original): The method of claim 2, wherein determining the second final parameter value includes evaluating an expression.

7. (Original): The method of claim 6, wherein the expression computes metadata.

8. (Original): The method of claim 2, wherein a prompt for receiving user input is conditional, and displaying the prompt depends upon evaluation of user input to a prior displayed prompt.

9. (Currently Amended): A method for modifying a graph at runtime execution of the graph, the graph having vertices representing components with parameters and links between components indicating flows of data between such components, the method including:

- (a) determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having a condition and a condition-interpretation;
- (b) evaluating the condition for every such conditional component; and
- (c) modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding condition-interpretation of such conditional component.

10. (Original): The method of claim 9, wherein modifying the graph includes removing the conditional component and all connected flows to such conditional component from the graph before execution of the graph based on a first evaluation of the condition and the corresponding condition-interpretation for such conditional component.

11. (Original): The method of claim 10, further including removing each component and flows connected to such components that depend on the presence of the conditional component.

12. (Original): The method of claim 9, wherein modifying the graph includes replacing the conditional component with a flow before execution of the graph based on a second evaluation of the condition and the corresponding condition-interpretation for such conditional component.

13. (Original): The method of claim 9, further including providing an interface which permits designating a condition and a condition-interpretation for a graph component.

14. (Currently Amended): A system for executing a graph having vertices representing components and links between components indicating flows of data between such components, the graph having components with parameters, including:

- (a) means for retrieving a runtime parameter for the graph at runtime execution of the graph, the runtime parameter having a value defined as determinable at runtime execution of the graph;
- (b) means for determining whether the value for the runtime parameter is to be provided by user input;
- (c) means for displaying a prompt to a user for receiving user input for every runtime parameter so determined;
- (d) means for determining a first final parameter value based on user response to such prompt; and
- (e) means for executing the graph using the first final parameter value as the value for the runtime parameter.

15. (Original): The system of claim 14, further including:

- (a) means for determining whether the value for the runtime parameter is to be externally supplied programmatically; and

- (b) means for retrieving any externally supplied value for every runtime parameter so determined;
- (c) means for determining a second final parameter value based on such externally supplied value; and
- (d) means for executing the graph using the second final parameter value as the value for the runtime parameter.

16. (Original): The system of claim 14, further including an interface which permits designating a parameter of a graph component as a runtime parameter.

17. (Original): The system of claim 14, wherein the means for determining the first final parameter value includes means for evaluating an expression.

18. (Original): The system of claim 17, wherein the expression computes metadata.

19. (Original): The system of claim 15, wherein the means for determining the second final parameter value includes means for evaluating an expression.

20. (Original): The system of claim 19, wherein the expression computes metadata.

21. (Original): The system of claim 15, wherein a prompt for receiving user input is conditional, and displaying the prompt depends upon evaluation of user input to a prior displayed prompt.

22. (Currently Amended): A system for modifying a graph at runtime execution of the graph, the graph having vertices representing components with parameters and links between components indicating flows of data between such components, the system including:

- (a) means for determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having a condition and a condition-interpretation;
- (b) means for evaluating the condition for every such conditional component; and
- (c) means for modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding condition-interpretation of such conditional component.

23. (Original): The system of claim 22, wherein the means for modifying the graph includes means for removing the conditional component and all connected flows to such conditional component from the graph before execution of the graph based on a first evaluation of the condition and the corresponding condition-interpretation for such conditional component.

24. (Original): The system of claim 23, further including means for removing each component and flows connected to such components that depend on the presence of the conditional component.

25. (Original): The system of claim 22, wherein the means for modifying the graph includes means for replacing the conditional component with a flow before execution of the graph based on a second evaluation of the condition and the corresponding condition-interpretation for such conditional component.

26. (Original): The system of claim 22, further including an interface which permits designating a condition and a condition-interpretation for a graph component.

27. (Currently Amended): A computer program, stored on a computer-readable medium, for executing a graph having vertices representing components and links between components indicating flows of data between such components, the graph having components with parameters, the computer program comprising instructions for causing a computer to:

- (a) retrieve a runtime parameter for the graph at runtime execution of the graph, the runtime parameter having a value defined as determinable at runtime execution of the graph;
- (b) determine whether the value for the runtime parameter is to be provided by user input;
- (c) display a prompt to a user for receiving user input for every runtime parameter so determined;
- (d) determine a first final parameter value based on user response to such prompt; and
- (e) execute the graph using the first final parameter value as the value for the runtime parameter.

28. (Original): The computer program of claim 27, further including instructions for causing the computer to:

- (a) determine whether the value for the runtime parameter is to be externally supplied programmatically; and
- (b) retrieve any externally supplied value for every runtime parameter so determined;
- (c) determine a second final parameter value based on such externally supplied value; and
- (d) execute the graph using the second final parameter value as the value for the runtime parameter.

29. (Original): The computer program of claim 27, further including instructions for causing the computer to provide an interface which permits designating a parameter of a graph component as a runtime parameter.

30. (Original): The computer program of claim 27, wherein the instructions for causing the computer to determine the first final parameter value include instructions for causing the computer to evaluating an expression.

31. (Original): The computer program of claim 30, wherein the expression computes metadata.

32. (Original): The computer program of claim 28, wherein the instructions for causing the computer to determine the second final parameter value include instructions for causing the computer to evaluating an expression.

33. (Original): The computer program of claim 32, wherein the expression computes metadata.

34. (Original): The computer program of claim 28, wherein a prompt for receiving user input is conditional, and displaying the prompt depends upon evaluation of user input to a prior displayed prompt.

35. (Currently Amended): A computer program, stored on a computer-readable medium, for modifying a graph at runtime execution of the graph, the graph having vertices representing components with parameters and links between components indicating flows of data between such components, the computer program comprising instructions for causing a computer to:

- (a) determine at runtime execution of the graph whether any component of the graph is defined as being a conditional component having a condition and a condition-interpretation;
- (b) evaluate the condition for every such conditional component; and
- (c) modify the graph at runtime execution of the graph in accordance with such evaluation and the corresponding condition-interpretation of such conditional component.

36. (Original): The computer program of claim 35, wherein the instructions for causing the computer to modify the graph include instructions for causing the computer to remove the conditional component and all connected flows to such conditional component from the graph before execution of the graph based on a first evaluation of the condition and the corresponding condition-interpretation for such conditional component.

37. (Original): The computer program of claim 36, further including instructions for causing the computer to remove each component and flows connected to such components that depend on the presence of the conditional component.

38. (Original): The computer program of claim 35, wherein the instructions for causing the computer to modify the graph include instructions for causing the computer to replace the conditional component with a flow before execution of the graph based on a second evaluation of the condition and the corresponding condition-interpretation for such conditional component.

39. (Original): The computer program of claim 35, further including instructions for causing the computer to provide an interface which permits designating a condition and a condition-interpretation for a graph component.